

نام دانشجو : بهاره غلامی

شماره دانشجویی: 40033626

نام استاد درس : دکتر مصطفی فخر احمد

موضوع : پیاده سازی الگوریتم a\_priori

الگوریتم :

الگوریتم اپریوری (Apriori) بر این اساس است که اگر یک مجموعه اقلام (itemset) مکرر (فرکانس بالا دارد) است، پس همه زیرمجموعه‌های آن نیز مکرر هستند. این بدین معنا است که اگر  $\{0,1\}$  دارای فرکانس بالا باشند، پس  $\{0\}$  و  $\{1\}$  نیز دارای فرکانس بالا هستند. بالعکس این قاعده نیز صادق است، یعنی اگر یک مجموعه اقلام دارای فرکانس بالا نباشد، زیرمجموعه‌های آن نیز دارای فرکانس بالا نیست. با توجه به توضیحات بالا، برای یافتن یک مجموعه قواعد وابستگی، ابتدا باید مجموعه اقلام مکرر را پیدا کنیم.

پیاده سازی الگوریتم :

```
def load_dataset()
```

برای لود کردن دیتاست از تابع بالا استفاده میکنیم .

```
def build_matrix1(Sequences):
```

ورودی این تابع تراکنش‌های ما هست که از دیتاست دریافت کرده ایم در این تابع ماتریسی به دست می‌آوریم که ماتریس به این صورت هست که ستون‌های آن شامل item ها است و سطر آن شامل تراکنش‌ها است و در صورتی که item ،  $i$ ام در تراکنش  $i$  وجود داشته باشد آن خانه را برابر یک می‌گذاریم .  
کار دیگری که این تابع میکند مجموعه item ها را نیز به دست می‌آورد .

```
def gnr_L1(matrix,min_sup)
```

در این تابع ما تعداد هر `item` را می‌شماریم و در صورتی که از منیموم ساپورت بیشتر بود آن تعداد `item` را نگه می‌داریم و در غیر این صورت آن را حذف می‌کنیم.

```
def gnrate_L(L1,min_sup,matrix,item)
```

سپس `item` های باقی مانده را با استفاده از تابع `super_set` با هم `join` می‌کنیم و سپس ترکیب دوتایی از `item` ها را محاسبه می‌کنیم و تعداد فرانس آن ها را با منیموم ساپورت مقایسه می‌کنیم.

و همین کار را ادامه می‌دهیم و ایت‌های دوتایی را با استفاده از `make_join` با هم `join` کرده و سپس ترکیب های سه تایی را به دست می‌آوریم و چک می‌کنیم که زیر مجموعه این ترکیب های سه تایی `frequent` باشند تا از اسکن های بیهوده جلوگیری کنیم تا سرعت الگوریتم بالاتر برود.

به عنوان مثال مجموعه `[a,b,c]` در صورتی که `[a,b]` که زیر مجموعه آن است `frequent` نباشد این مجموعه سه تایی را حذف می‌کنیم.

سپس فرکانس این ترکیب های سه تایی را محاسبه و با منیموم ساپورت مقایسه می‌کنیم و در صورتی که فرکانس کمتر از منیموم ساپورت باشد آن ها را هرس می‌کنیم.

این مراحل تا زمانی ادامه می‌دهیم که نتوانیم از خروجی `make_join` زیر مجموعه `n` تایی بسازیم و به تهی برسیم.

خروجی تابع `frequent_itemset` ها است که مربوط به قسمت اول تمرین است و آن ها در فایل `frequent_itemset.txt` ذخیره کردیم.

قسمتی از نتایج :

```
EX_3.py  frequent_itemset.txt x
frequent_itemset.txt
1 frequent_itemset 2:
2 [Left 4 Dead,Minecraft,]
3 [Left 4 Dead,The Last of Us,]
4 [Left 4 Dead,Grand Theft Auto V,]
5 [Left 4 Dead,Read Dead Redemption,]
6 [Left 4 Dead,God of War,]
7 [Minecraft,The Last of Us,]
8 [Minecraft,Grand Theft Auto V,]
9 [Minecraft,Read Dead Redemption,]
10 [Minecraft,God of War,]
11 [The Last of Us,Grand Theft Auto V,]
12 [Read Dead Redemption,The Last of Us,]
13 [The Last of Us,God of War,]
14 [Read Dead Redemption,Grand Theft Auto V,]
15 [God of War,Grand Theft Auto V,]
16 [Read Dead Redemption,God of War,]
17 [Left 4 Dead,Assassin's Creed 2,]
18 [Assassin's Creed 2,Read Dead Redemption,]
19 [Assassin's Creed 2,God of War,]
20 [Left 4 Dead,The Elder Scrolls V: Skyrim,]
21 [Left 4 Dead,Super Mario Bros 3,]
```

```
frequent_itemset.txt - Notepad
File Edit Format View Help
frequent_itemset 2:
[The Last of Us,Minecraft,]
[The Last of Us,God of War,]
[The Last of Us,Left 4 Dead,]
[Grand Theft Auto V,The Last of Us,]
[Read Dead Redemption,The Last of Us,]
[Minecraft,god of war,]
[Minecraft,Left 4 Dead,]
[Grand Theft Auto V,Minecraft,]
[Read Dead Redemption,Minecraft,]
[God of War,Left 4 Dead,]
[Grand Theft Auto V,God of War,]
[Read Dead Redemption,God of War,]
[Grand Theft Auto V,Left 4 Dead,]
[Read Dead Redemption,Left 4 Dead,]
[Grand Theft Auto V,Read Dead Redemption,]
[Assassin's Creed 2,God of War,]
[Assassin's Creed 2,Left 4 Dead,]
[Assassin's Creed 2,Read Dead Redemption,]
[Assassin's Creed 2,Super Mario World,]
[Assassin's Creed 2,The Last of Us,]
[Assassin's Creed 2,The Elder Scrolls V: Skyrim,]
[Super Mario World,The Last of Us,]
[Super Mario World,Left 4 Dead,]
[Read Dead Redemption,Super Mario World,]
[Super Mario World,The Elder Scrolls V: Skyrim,]
[The Elder Scrolls V: Skyrim,The Last of Us,]
[The Elder Scrolls V: Skyrim,Left 4 Dead,]
[Read Dead Redemption,The Elder Scrolls V: Skyrim,]
[Resident Evil 4,The Last of Us,]
[Resident Evil 4,Minecraft,]
[Resident Evil 4,Left 4 Dead,]
[Read Dead Redemption,Resident Evil 4,]
[Resident Evil 4,Dark Souls,]
[The Last of Us,Dark Souls,]
[Dark Souls,Minecraft,]
[Dark Souls,Left 4 Dead,]
[Read Dead Redemption,Dark Souls,]
[Assassin's Creed 2,Resident Evil 4,]
[Grand Theft Auto V,Assassin's Creed 2,]
[Assassin's Creed 2,Guitar Hero 3,]
[Assassin's Creed 2,Dark Souls,]
[Super Mario World,Resident Evil 4,]
Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

```
def frequency(sub_set, matrix)
```

با استفاده از تابع بالا فرکانس مجموعه اِیتم های دو به بالا را محاسبه میکنم .

```
def findsubsets(s, n)
```

از این تابع برای محاسبه زیرمجموعه با تعداد عضو n استفاده میکنیم .

```
def subtract (l1,l2)
```

با استفاده از تابع بالا دو مجموعه را از هم تفریق میکنیم .

```
def generate_rules(Map,L,min_conf,size_matrix,item)
```

این قسمت مربوط به قسمت دوم تمرین است که میخواهیم قوانین با درجه اعتماد بالای 50 درصد را پیدا کنیم.

بعد از به دست آمدن لسیت ها و مجموعه item های پرتکرار را به این تابع می‌دهیم تا باتوجه به confidence قواعد که قوی را پیدا کنیم با استفاده از لیستی که شامل l2 تا lk است (با استفاده از تابع gnrate\_L) زیر مجموعه ان ها (سمت چپ) را محاسبه کرده و از مجموعه اصلی کم میکنیم (سمت راست قانون) و با confidence مقایسه میکنیم تا قوانین را به دست آوریم و سپس با استفاده و قوانین را بر اساس ساپورت مرتب میکنیم .  
مجموعا 1909 قانون به دست می آید.

```
def get_txtfile(result)
```

با استفاده از این تابع قوانین به دست آمده در فایل result.txt ذخیره میکنیم .

قسمتی از نتایج قانون های به دست اومده:



```
1:['Left 4 Dead', 'Minecraft', 'Dark Souls', 'The Last of Us', 'The Elder Scrolls V: Skyrim', 'Guitar Hero 3', 'Assassin's Creed 2', 'Super Mario World']----->['Read Dead Redemption'] Support : 0.10378412901165576 Confidence : 50.0
2:['Left 4 Dead', 'Minecraft', 'Dark Souls', 'The Last of Us', 'The Elder Scrolls V: Skyrim', 'Guitar Hero 3', 'Assassin's Creed 2', 'Read Dead Redemption']----->['Super Mario World'] Support : 0.10378412901165576 Confidence : 50.0
3:['Left 4 Dead', 'Minecraft', 'Dark Souls', 'The Last of Us', 'The Elder Scrolls V: Skyrim', 'Guitar Hero 3', 'Super Mario World', 'Read Dead Redemption']----->['Assassin's Creed 2'] Support : 0.10378412901165576 Confidence : 52.0
4:['Left 4 Dead', 'Minecraft', 'Dark Souls', 'The Last of Us', 'The Elder Scrolls V: Skyrim', 'Assassin's Creed 2', 'Super Mario World', 'Read Dead Redemption']----->['Guitar Hero 3'] Support : 0.10378412901165576 Confidence : 56.52173913043478
5:['Left 4 Dead', 'Minecraft', 'Dark Souls', 'The Elder Scrolls V: Skyrim', 'Guitar Hero 3', 'Assassin's Creed 2', 'Super Mario World', 'Read Dead Redemption']----->['The Last of Us'] Support : 0.10378412901165576 Confidence : 72.22222222222221
6:['Left 4 Dead', 'Minecraft', 'The Last of Us', 'The Elder Scrolls V: Skyrim', 'Guitar Hero 3', 'Assassin's Creed 2', 'Super Mario World', 'Read Dead Redemption']----->['Dark Souls'] Support : 0.10378412901165576 Confidence : 54.166666666666664
7:['Left 4 Dead', 'Dark Souls', 'The Last of Us', 'The Elder Scrolls V: Skyrim', 'Guitar Hero 3', 'Assassin's Creed 2', 'Super Mario World', 'Read Dead Redemption']----->['Minecraft'] Support : 0.10378412901165576 Confidence : 54.166666666666664
8:['Minecraft', 'Dark Souls', 'The Last of Us', 'The Elder Scrolls V: Skyrim', 'Guitar Hero 3', 'Assassin's Creed 2', 'Super Mario World', 'Read Dead Redemption']----->['Left 4 Dead'] Support : 0.10378412901165576 Confidence : 52.0
9:['Left 4 Dead', 'Minecraft', 'Dark Souls', 'The Elder Scrolls V: Skyrim', 'Guitar Hero 3', 'Grand Theft Auto V', 'Super Mario World', 'Read Dead Redemption']----->['God of War'] Support : 0.10378412901165576 Confidence : 52.0
10:['Left 4 Dead', 'Minecraft', 'Dark Souls', 'The Elder Scrolls V: Skyrim', 'Guitar Hero 3', 'Super Mario World', 'Read Dead Redemption', 'God of War']----->['Grand Theft Auto V'] Support : 0.10378412901165576 Confidence : 52.0
11:['Left 4 Dead', 'Minecraft', 'Dark Souls', 'The Elder Scrolls V: Skyrim', 'Grand Theft Auto V', 'Super Mario World', 'Read Dead Redemption', 'God of War']----->['Guitar Hero 3'] Support : 0.10378412901165576 Confidence : 56.52173913043478
12:['Left 4 Dead', 'Minecraft', 'Dark Souls', 'Guitar Hero 3', 'Grand Theft Auto V', 'Super Mario World', 'Read Dead Redemption', 'God of War']----->['The Elder Scrolls V: Skyrim'] Support : 0.10378412901165576 Confidence : 54.166666666666664
13:['Left 4 Dead', 'Minecraft', 'The Elder Scrolls V: Skyrim', 'Guitar Hero 3', 'Grand Theft Auto V', 'Super Mario World', 'Read Dead Redemption', 'God of War']----->['Dark Souls'] Support : 0.10378412901165576 Confidence : 68.42105263157895
14:['Minecraft', 'Dark Souls', 'The Last of Us', 'The Elder Scrolls V: Skyrim', 'Grand Theft Auto V', 'Assassin's Creed 2', 'Super Mario World', 'God of War']----->['Read Dead Redemption'] Support : 0.10378412901165576 Confidence : 54.166666666666664
15:['Minecraft', 'Dark Souls', 'The Last of Us', 'The Elder Scrolls V: Skyrim', 'Assassin's Creed 2', 'Super Mario World', 'Read Dead Redemption', 'God of War']----->['Grand Theft Auto V'] Support : 0.10378412901165576 Confidence : 61.904761904761905
16:['Minecraft', 'Dark Souls', 'The Last of Us', 'Grand Theft Auto V', 'Assassin's Creed 2', 'Super Mario World', 'Read Dead Redemption', 'God of War']----->['The Elder Scrolls V: Skyrim'] Support : 0.10378412901165576 Confidence : 52.0
17:['Minecraft', 'Dark Souls', 'The Elder Scrolls V: Skyrim', 'Grand Theft Auto V', 'Assassin's Creed 2', 'Super Mario World', 'Read Dead Redemption', 'God of War']----->['The Last of Us'] Support : 0.10378412901165576 Confidence : 56.52173913043478
18:['Minecraft', 'The Last of Us', 'The Elder Scrolls V: Skyrim', 'Grand Theft Auto V', 'Assassin's Creed 2', 'Super Mario World', 'Read Dead Redemption', 'God of War']----->['Dark Souls'] Support : 0.10378412901165576 Confidence : 50.0
19:['Dark Souls', 'The Last of Us', 'The Elder Scrolls V: Skyrim', 'Guitar Hero 3', 'Grand Theft Auto V', 'Resident Evil 4', 'Assassin's Creed 2', 'Read Dead Redemption']----->['God of War'] Support : 0.10378412901165576 Confidence : 50.0
20:['Dark Souls', 'The Last of Us', 'The Elder Scrolls V: Skyrim', 'Guitar Hero 3', 'Grand Theft Auto V', 'Resident Evil 4', 'Read Dead Redemption', 'God of War']----->['Assassin's Creed 2'] Support : 0.10378412901165576 Confidence : 56.52173913043478
21:['Dark Souls', 'The Last of Us', 'The Elder Scrolls V: Skyrim', 'Guitar Hero 3', 'Resident Evil 4', 'Assassin's Creed 2', 'Read Dead Redemption', 'God of War']----->['Grand Theft Auto V'] Support : 0.10378412901165576 Confidence : 54.166666666666664
22:['Dark Souls', 'The Last of Us', 'The Elder Scrolls V: Skyrim', 'Grand Theft Auto V', 'Resident Evil 4', 'Assassin's Creed 2', 'Read Dead Redemption', 'God of War']----->['Guitar Hero 3'] Support : 0.10378412901165576 Confidence : 54.166666666666664
```

و مجموعا تعداد 1909 تا قانون به دست می آوریم.

```
def apriori(min_sup,min_conf)
```

برای فراخوانی توابع نوشته شده و مرحله بندی الگوریتم استفاده میکنیم از این تابع که ابتدا دیتاست را لود میکند و سپس ماتریس و مجموعه ایتm ها به دست می اوریم و سپس مجموعه l1 تا lk را به دست می اوریم که پرتکرار هستند را به دست می اوریم و در فایل ذخیره میکنیم و از روی انها قوانین را حساب میکنم و در فایل ذخیره میکنیم .

**def recomment(role,item):**

قسمت سوم تمرین مربوط به سیستم پیشنهاد دهنده است که ان را با تابع بالا پیاده سازی میکنیم.

به این صورت عمل میکند که لیست خرید های مشتری را میگیرد و با سمت چپ قوانین مقایسه میکند در صورتی که یکسان باشد قسمت راست قانون را پیشنهاد میدهد به مشتری .

نتیجه:

```

1897 1897:['The Elder Scrolls V: Skyrim', 'God of War', 'Super Mario World']----->['Resident
1898 1898:['The Elder Scrolls V: Skyrim', 'Resident Evil 4', 'Super Mario World']----->['God
1899 1899:['God of War', 'Resident Evil 4', 'Guitar Hero 3']----->['Super Mario World'] Suppo
1900 1900:['Super Mario World', 'Resident Evil 4', 'Guitar Hero 3']----->['God of War'] Suppo
1901 1901:['God of War', 'Super Mario World', 'Guitar Hero 3']----->['Resident Evil 4'] Suppo
1902 1902:['God of War', 'Dark Souls', 'Resident Evil 4']----->['Read Dead Redemption'] Suppo
1903 1903:['God of War', 'Dark Souls', 'Read Dead Redemption']----->['Resident Evil 4'] Suppo
1904 1904:['God of War', 'Resident Evil 4', "Assassin's Creed 2"]----->['Super Mario World']
1905 1905:['Super Mario World', 'Resident Evil 4', "Assassin's Creed 2"]----->['God of War']
1906 1906:['God of War', 'Super Mario World', "Assassin's Creed 2"]----->['Resident Evil 4']
1907 1907:['Dark Souls', 'Resident Evil 4']----->['Read Dead Redemption'] Support : 9.9712597
1908 1908:['Super Mario World', 'Resident Evil 4']----->['God of War'] Support : 10.098994092
1909 1909:['God of War', 'Super Mario World']----->['Resident Evil 4'] Support : 10.098994092
1910

```

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

pleas select one number of item:
2
pleas select one number of item:
stop
select by client:
['God of War', 'Super Mario World']
New Game Recommendations :
['Resident Evil 4']

```

همان طور که مشخص است در صورتی که کاربر ایتm هایی که وارد میکند با سمت چپ قانون 1909 یکی باشد انگاه ما به ان سمت راست قانون را پیشنهاد میدهیم.

